*provisional date good enough*

PGPUB-DOCUMENT-NUMBER:    20040068476

PGPUB-FILING-TYPE:        new

DOCUMENT-IDENTIFIER:      US 20040068476 A1

TITLE:                    System, process and software arrangement for assisting
                          with a knowledge discovery

PUBLICATION-DATE:         April 8, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY |
|------|------|-------|---------|
| RULE-47 | | | |
| Provost, Foster | New York | NY | US |
| Bernstein, Abraham | New York | NY | US |

US-CL-CURRENT:    706/45

ABSTRACT:

A process, system and computer software are provided to produce at least two
solutions related to a knowledge discovery from data.  In particular, the
information regarding operators which are usable for the knowledge discovery of
the data is received, and the solutions are generated.  Each of the solutions
includes at least one of the operators.  An ability is provided to select at
least one of the solutions so as to execute one or more procedures from the
data.  Each of the procedures is associated with the operator of the respective
solution.  In addition, it is possible to include a variable number of the
operators in at least one (and possibly all) of the solutions.  Also, it is
possible to generate a code for at least one automatically-generated solution.
In particular, access to the automatically-generated solution can be obtained
with this solution that includes operators usable for the knowledge discovery
of the data.  Then, the code can be generated for associating one of the
operators of the automatically-generated solution with another one of the
operators of the automatically-generated solution.

---------- KWIC ---------


Detail Description Paragraph - DETX (19):
    [0063] In addition, the ontology preferably groups the KD
procedures/operators into logical groups, which can be used to narrow the set
of procedures/operators to be considered at each stage in the KD process.  FIG.
5 shows a exemplary overall tree-type structure of an ontology which groups the
KD procedures/operators into three groups, and which can be used with the
system of FIG. 3 and the process of FIGS. 4A and 4B according to the present
invention.  These three groups are, e.g., a pre-processing group, an induction
algorithm group, and a post-processing group.  As illustrated in FIG. 5, each
of these groups is further subdivided.  In particular, the leafs of the
tree-type structure of the ontology according to an exemplary embodiment of the
present invention are preferably the procedures/operators.  For example, the
induction algorithm group can be subdivided into classifiers, class probability
estimators ("CPEs") and regressors.  The classifiers can further be grouped
into decision trees (e.g., a C4.5 algorithm as described in J.R.  Quinlan,
"C4.5: Programs for Machine Learning", San Mateo, Calif., Morgan Kaufmann,
1993) and rule learners (e.g., a "PART" algorithm as described in E. Frank et
al., "Generating Accurate Rule Sets Without Global Optimization", In
Proceedings of the Fifteenth International Conference on Machine Learning,

1998, pp. 144-151).


Detail Description Paragraph - DETX (24):
    [0068] In step 420, the goal-state information (e.g., the goal criteria) can
be obtained, e.g., from the user via user interface 425 by using graphical user
interface dialog boxes.  It is also possible for the processing device 100 to
utilize default goal-state information which can be inferred from the meta
data.  A variety of the goal-state information can be obtained which may be,
e.g., a high accuracy, fast mining, fast model execution, comprehensible
output, cost-sensitive operation, few features used,. etc. It should be
understood that these criteria may at times be contradictory to one another.
In such situations, the processing device 100 may provide to the user a method
for specifying the desired tradeoffs.  For example, if accuracy and speed of
learning are incompatible for particular data, the processing device 100 may
display to the user a slide bar or a weighting **scheme** for specifying which is
more important to the user, and to what extent.  In one exemplary embodiment,
the user may be requested to provide the structure of the desired model, e.g.,
format (decision tree, rule-set, equation, etc.), type (class probability
estimator, classifier, regressor, etc.), comprehensibility (vocabulary, model,
etc.), size, cost sensitivity, speed, and/or others.


Detail Description Paragraph - DETX (33):
    [0077] In still another exemplary embodiment of the present invention, each
of the valid KD processes/plans/solutions is executed on the test data to
provide the score based on accuracy of the results provided by each respective
plan/solution.  The accuracy can be determined by **comparing** the results of each
determination (using the associated plan/solution) to the real results (which
were previously calculated).  In this manner, the processing device 100 and the
IDEA can rank the valid KD processes/plans/solutions based on the accuracy of
the results of each.  Criteria other than the accuracy and speed can also be
used, and are within the scope of the present invention.


Detail Description Paragraph - DETX (39):
    [0083] In yet another exemplary embodiment of the present invention, it is
preferable to add a template procedure to the **ontology** of the exemplary
embodiment of the IDEA for a particular application (e.g., target marketing)
which defines a structure on the IDEA.  Various types of models can be used,
e.g., for the **class**-probability estimation and the regression model.  The
template would impose limitations on the IDEA as to which **classes** of operators
are to be executed at particular points of the execution of the knowledge
discovery process.  It is possible to utilize a default template which has
predetermined restrictions that should be imposed on the IDEA.  In addition,
the template can provide an ability to execute two or more of the KD processes
(or a set of processes) simultaneously.


Detail Description Paragraph - DETX (47):
    [0091] FIG. 15 shows a detailed illustration 700 for plan # 90.  As shown in
FIG. 14, this plan first randomly samples a subset of the data (smaller data
size may lead to a faster induction).  Then, it applies fixed-bin
discretization to transform the numeric variables into categorical variables.
It should be noted that the C4.5 procedure does not require a discretization as
a pre-process.  However the **ontology** specifies that discretization can
accelerate the induction algorithms, and that the fixed-bin discretization is
generally faster than **class**-based discretization (e.g., the C4.5 procedure is
generally much faster without being applied on numeric variables, especially on
large data sets).  Thereafter, plan # 90 utilizes feature sampling.  The
**ontology** specified that the feature sampling may decrease accuracy, but will
provide a large increase in speed.  The KD process planner would make the
tradeoff due to the user's instructions.  Finally, the IDEA would use the C4.5
procedure, which the **ontology** specified to be a fast learning algorithm.

Thereafter, in FIG. 16, the user can select any one or more of the valid KD processes/plans/solutions using an export executable plan interface (which can generate computer code), and possibly record the selected one or more plans/solutions using a record interface 750 to a file.